

具有复杂约束混料试验的渐近D-最优设计 *

李光辉 张崇岐*

(广州大学经济与统计学院, 广州, 510006)

摘要: 复杂约束条件下试验设计区域极不规则, 通常难以得到精确的最优设计. 本文构造一种针对混料试验设计的随机搜索算法(MDRS), 在具有复杂约束的区域内由Monte-Carlo方法产生一组初始点集, 并通过MDRS算法迭代至逼近最优点集. 通过实例验证, 这种方法是有效的. 它可以作为衡量其他设计的一个标准, 即只有当给出的其他设计优于近似的最优解时才是有效.

关键词: Monte-Carlo方法; D-准则; 信息矩阵

中图分类号: O212.6

英文引用格式: Li G H, Zhang C Q. Asymptotically D-optimal design of mixture experiment with complex constraints [J]. Chinese J. Appl. Probab. Statist., 2017, 33(2): 203–220. (in Chinese)

§1. 引言

自上世纪五十年代末, Scheffé首次提出混料试验设计的概念以来, 该方向的研究发展至今主要有两类设计. 一类是在各种最优准则的基础上提出的设计. 例如Scheffé多项式模型及相应的单纯形格子设计和单纯形中心设计^[1], Cox设计和轴设计等^[2]. 常用的最优设计准则有D-最优准则、G-最优准则、A-最优准则等等^[3], 其中D-准则是应用最为广泛的一种评价设计最优性的准则. 另一类设计是在均匀性和稳健性的前提下提出的混料均匀设计^[4]. 文献[5]通过特殊变换超级立方体中的均匀性较好的设计得到混料均匀设计, 此后许多文献推广发展了这一方法^[6,7]. 这类设计是将混料试验点均匀的安排在试验域内, 它对模型的假设较少, 因而更具有普遍性和稳健性.

两类设计各有优势. 一般的, 如果真实模型的形式已知, 采用最优设计所建立的模型与真实模型会更接近, 其预测效果也会更优; 反之如果试验者并不知道真实模型的形式, 则均匀设计更为稳健. 在实际应用中, 混料试验域通常还含有其它给定的附加约束条件, 这样就使得试验区域变得更为复杂. 近年来, 对于不规则区域上填充设计的问题引起统计学家和实际工作者的重视.

对于 q 分量的混料系统, 响应是各分量比例 x_1, x_2, \dots, x_q 的函数, 由各分量比例所确定的 $q-1$ 维单纯形可表示为

$$S^{q-1} = \left\{ (x_1, x_2, \dots, x_q) : \sum_{i=1}^q x_i = 1, x_i \geq 0, i = 1, 2, \dots, q \right\}.$$

*国家自然科学基金项目(批准号: 11671104)资助.

*通讯作者, E-mail: cqzhang@gzhu.edu.cn.

本文2015年12月18日收到, 2016年3月11日收到修改稿.

如果模型还含有其它的约束条件, 试验域可表示为

$$\mathcal{D}_s = \left\{ (x_1, x_2, \dots, x_q) : \sum_{i=1}^q x_i = 1, x_i \geq 0, i = 1, 2, \dots, q, C's \right\},$$

其中 $C's$ 表示附加约束条件.

在混料试验中, 附加约束条件主要有三种形式.

- (i) 兼有上、下界约束的条件: $C's : 0 \leq a_j \leq x_j \leq b_j \leq 1, j = 1, 2, \dots, q;$
- (ii) 线性约束条件: $C's : 0 \leq a_j \leq \sum_{i=1}^q c_{ji} x_i \leq b_j \leq 1, j = 1, 2, \dots, m;$
- (iii) 非线性约束条件: $C's : a_j \leq \phi_j(x_1, x_2, \dots, x_q) \leq b_j \leq 1, j = 1, 2, \dots, l;$

其中 a_j, b_j 都是已知的常数, $\phi_j(x_1, x_2, \dots, x_q)$ 是各变量的非线性函数.

当模型含有非线性约束条件时, 区域 \mathcal{D}_s 形状极不规则. 基于均匀设计的思想, Liu 提出一种方法, 能在含有复杂约束的混料区域上构造均匀设计点, 并且给出了实例^[8]. 从最优设计的角度来看, 如果混料模型只含有线性约束或上、下界约束条件, 基于信息矩阵的各类最优化准则下, 最优设计点只可能是某些极端顶点, 约束边界中心以及棱中心所组成的集合. 文献[11]中介绍了兼有上、下界约束的极端顶点和约束边界中心以及棱中心的计算方法. 但含有非线性约束的区域 \mathcal{D}_s 有可能不存在极端顶点与棱, 或者所有的极端顶点或棱中心组成的点集并不一定是最优设计点. 目前关于约束区域内的最优设计的研究还大多是讨论选择极端顶点及边界点的组合问题.

鉴于此, 本文构造一种适用于混料试验设计中搜索 D-最优设计的算法, 即 MDRS (mixture design random search) 算法. 该算法主要步骤是: 首先由 Monte-Carlo 模拟的方法产生一组初值, 然后运行搜索迭代, 在每一步迭代中产生若干随机点, 通过比较确定新的迭代点集. 使用 MDRS 算法可以搜索出具有复杂约束条件下的混料试验的近似 D-最优设计解, 这组解可以作为衡量试验设计的一个标准, 即只有当给出的其他设计优于近似的最优解时, 才是有效. 本文首先提出模型的基本假设和几个常用的公式; 第 3 节介绍了 MDRS 算法初值的产生和下界约束随机点的生成; 第 4 节介绍了在复杂区域上 D-最优的随机搜索方法, 即 MDRS 算法的迭代搜索过程; 第 5 节讨论加点试验和边界搜索; 最后总结和讨论可进一步研究的问题.

§2. 模型的基本假设与 D-最优准则

假设我们考虑的一般线性模型的基本形式为

$$y = \beta^\top f(x) + \varepsilon,$$

其中 y 是响应变量, $f(x) = (f_1(x), f_2(x), \dots, f_m(x))^\top$ 是给定的关于 $x = (x_1, x_2, \dots, x_q) \in \mathcal{X}$ 的函数向量, 试验域 \mathcal{X} 为拓扑空间上的紧集, $\beta = (\beta_1, \beta_2, \dots, \beta_m)^\top$ 是未知的参数向量, ε 是随机误差, 通常还假设 $E(\varepsilon) = 0, \text{Var}(\varepsilon) = \sigma^2$.

记 $\tau_j = (x_{j1}, x_{j2}, \dots, x_{jq}) \in \mathcal{X}$, $j = 1, 2, \dots, N$ 是试验域 \mathcal{X} 中的 N 个试验点. 记 $y = (y_1, y_2, \dots, y_N)^\top$ 为 N 次试验的结果. 若 N 次试验共有 k 个不同的设计点, 记为 $\tau_1, \tau_2, \dots, \tau_k$. 假设在试验点 τ_i 的测度为 w_i , 记

$$\xi = \begin{pmatrix} \tau_1 & \tau_2 & \dots & \tau_k \\ w_1 & w_2 & \dots & w_k \end{pmatrix}.$$

称 $\xi \in \Xi$ 为一个具有测度的 k 点设计, Ξ 称为设计空间. 定义信息矩阵为

$$M(\xi) = \int_{\mathcal{X}} f(x) f^\top(x) \xi(dx).$$

当 $\det(M(\xi)) \neq 0$ 时, 称 ξ 为一个非奇异的设计, 一般考虑的设计都假定为非奇异的. 如果回归模型的误差服从正态分布, 则 β 的最小二乘估计 $\hat{\beta}$ 具有正态性, 即有 $\hat{\beta} \sim N(\beta, \sigma^2 M^{-1}(\xi))$, 对于给定的置信水平 α 可以确定一个常数 $c(\alpha)$, 使得 β 的置信椭球可以表示为

$$\{\beta : (\beta - \hat{\beta})^\top M^{-1}(\xi) (\beta - \hat{\beta}) \leq c(\alpha)\}.$$

D-最优设计的原理是将信息矩阵的行列式最大化, 从而使得参数向量 β 的置信椭球体积最小. D-最优准则可表述为

$$\det(M(\xi)) \rightarrow \sup_{\xi \in \Xi} .$$

G-A等价定理是用于判断一个设计是否达到D-最优的重要准则. 对于任意的 $\tau \in \mathcal{X}$, 称关于 τ 的函数 $\delta(\tau, \xi) = f^\top(\tau) M^{-1}(\xi) f(\tau)$ 为方差函数, 设计 ξ 是D-最优设计的充分必要条件是

$$\max_{x \in \mathcal{X}} \delta(x, \xi) = m,$$

即使得方差函数的最大值等于回归方程中未知参数的个数.

为了后面讨论的需要, 在此我们给出点到点集的距离公式. 令 $A, B \subset \mathcal{X}$, 对于任意的 $x \in A$, 定义点 x 到 B 的距离为

$$d(x, B) = \inf\{d(x, y) : y \in B\},$$

其中 $d(x, y) = [(x - y)(x - y)^\top]^{1/2}$.

§3. 初值的产生与下界约束随机点集

为了确定最优设计 ξ^* , 一方面需要确定最优设计点集, 另一方面需要确定该点集中各点所对应的测度. 使用MDRS算法主要有三个步骤: 首先在复杂约束区域 \mathcal{D}_s 上生成一组点集作为随机搜索的初值, 其次运行随机搜索迭代, 最后对所得到的结果进行调整, 或者将结果再作为初值进行下一轮迭代, 以此来判断这组点集是否合适. 本节主要介绍产生初值的方法以及构造下界约束随机混料点集.

3.1 初值的产生

对于一个 k 点设计, 在随机搜索前需要产生一组初值 $\mathcal{T}^{(0)} = \{\tau_1^{(0)}, \tau_2^{(0)}, \dots, \tau_k^{(0)}\} \subset \mathcal{D}_s$, 并且使得 $\sum_{j=1}^k d(\tau_j^{(0)}, \mathcal{T}^*)$ 尽可能的小, 即要使得初始点集 $\mathcal{T}^{(0)}$ 与最优点集 \mathcal{T}^* 尽可能的接近. 设初值所对应的设计为 $\xi^{(0)}$, 因此我们需要选择的初值应使得 $\det M(\xi^{(0)})$ 尽可能的大. 为计算方便, 我们先规定各设计点测度相等, 在计算信息阵行列式时我们总是令 $\text{diag}(w_1, w_2, \dots, w_k) = \mathbf{I}_k$, \mathbf{I}_k 是 k 阶单位阵, 确定点集以后再对测度进行调整.

下面首先使用 Monte-Carlo 方法生成随机点集, 再从中抽取若干个 Bootstrap 随机样本, 经过比较可以确定初值点集, 具体过程如下.

生成一个随机矩阵 $Y = \{y_{ij}\}_{i,j=1}^{N,q}$, 这里要求 $N \gg k$, 其中 $y_{ij} \sim U(0, 1)$, $i = 1, 2, \dots, N$, $j = 1, 2, \dots, q$, 且各 y_{ij} 间相互独立. 记矩阵 $T = \{x_{ij}\}_{i,j=1}^{N,q}$ 是由矩阵 Y 通过逆变换而得到. 可以按指数逆变换法^[9], 定义 T 中的元素为

$$x_{ij} = \frac{\ln(1 - y_{ij})}{\sum_{j=1}^q \ln(1 - y_{ij})}, \quad i = 1, 2, \dots, N, j = 1, 2, \dots, q. \quad (1)$$

或者按照文献[10]的逆变换法, 令

$$\begin{cases} x_{ij} = (1 - y_{ij}^{1/(q-j)}) \prod_{l=1}^{j-1} y_{il}^{1/(q-l)}, & j = 1, 2, \dots, q-1; \\ x_{iq} = \prod_{l=1}^{q-1} y_{il}^{1/(q-l)}, & i = 1, 2, \dots, N. \end{cases} \quad (2)$$

记 $\tau_i = (x_{i1}, x_{i2}, \dots, x_{iq})$ 是矩阵 T 的第 i 行元素. 由(1)或(2)通过变换后所得到的矩阵 T , 其各行元素均满足条件 $\sum_{j=1}^q x_{ij} = 1$, $\tau_i \sim U(S^{q-1})$, $i = 1, 2, \dots, N$, 并且 $\tau_1, \tau_2, \dots, \tau_N$ 是相互独立的. 矩阵 T 中的各行元素 $\tau_1, \tau_2, \dots, \tau_N$ 都是 S^{q-1} 上的随机混料试验点. 记 $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_N\} \subset S^{q-1}$ 表示由矩阵 T 的各行元素作为试验点所组成的集合.

对于单纯形上具有复杂约束的区域 \mathcal{D}_s , 若 \mathcal{T} 中恰好有 n 个点落入 \mathcal{D}_s 内, 将这 n 个点组成的集合记为

$$\mathcal{T}_0 = \{\tau_1, \tau_2, \dots, \tau_n\} = \mathcal{T} \cap \mathcal{D}_s, \quad n \leq N.$$

将 \mathcal{T}_0 中各个点按行排列成的矩阵记为 $T_0 = (\tau_1^\top, \tau_2^\top, \dots, \tau_n^\top)^\top = \{x_{ij}\}_{i,j=1}^{n,q}$, 当 N 较大时, 落入区域 \mathcal{D}_s 内随机点的个数 n 有近似关系

$$n \approx N \frac{\text{Vol}(\mathcal{D}_s)}{\text{Vol}(S^{q-1})},$$

其中 $\text{Vol}(x)$ 表示区域 x 的体积, 且有 $\text{Vol}(S^{q-1}) = \sqrt{q}/(q-1)!$, 所以当 N 足够大时就能保证落入区域 \mathcal{D}_s 的随机混料点足够多.

当 n 很大时, 我们希望能从这 n 个点中找到一组点集 $\{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}\}$, $\{i_1, i_2, \dots, i_k\} \subset \{1, 2, \dots, n\}$, 使得在等测度条件下对应的设计的信息阵行列式达到最大. 若要进行全面比较, 则运算量很大不易实现, 并且, 我们需要确定的只是一组初值, 只要保证这组点集“较好”; 即可. 我们采用比较随机样本的方法进行选取.

在点集 \mathcal{T}_0 中随机产生 n 个Bootstrap样本, 记为 $\mathcal{T}_i = \{\tau_{i_1}, \tau_{i_2}, \dots, \tau_{i_k}\} \subset \mathcal{T}_0$, $i = 1, 2, \dots, n$, 记 \mathcal{T}_i 对应的设计为 ξ_i , 在等测度条件下计算出各个 $\det M(\xi_i)$, $i = 1, 2, \dots, n$ 并选出其中的最大者, 记为

$$\xi^{(0)} = \arg \max_{1 \leq i \leq n} \det M(\xi_i).$$

$\xi^{(0)}$ 所对应的试验点集记为 $\mathcal{T}^{(0)} = \{\tau_1^{(0)}, \tau_2^{(0)}, \dots, \tau_k^{(0)}\} \subset \mathcal{D}_s$, 以此作为随机搜索迭代的初值. 当然, 虽然初值的选取比较重要, 但从大量Bootstrap样本中选出的最优者一般都能确定一组较好的初值.

3.2 构造下界约束的混料试验点

设各个分量 x_1, x_2, \dots, x_q 的下界约束为 a_1, a_2, \dots, a_q , 由此所确定的具有下界约束的 $q - 1$ 维单纯形可表示为

$$S_a^{q-1} = \left\{ (x_1, x_2, \dots, x_q) : \sum_{i=1}^q x_i = 1, 0 \leq a_i \leq x_i \leq 1, i = 1, 2, \dots, q \right\}. \quad (3)$$

S_a^{q-1} 不是一点或是空集的充要条件是 $\sum_{i=1}^q a_i < 1$, 且各分量还应具有隐上界约束: $0 \leq a_i \leq x_i \leq 1 - \sum_{j \neq i} a_j$, 则称 S_a^{q-1} 为非退化的具有下界约束的混料试验设计域. 设 A 是单纯形 S_a^{q-1} 中的任意一点, 我们可以通过线性变换的方式将 S_a^{q-1} 中的自然分量坐标转换为无附加约束的标准混料单纯形中的坐标. 记点 A 的自然分量坐标及拟分量坐标分别为

$$\begin{aligned} x(A) &= (x_1(A), x_2(A), \dots, x_q(A)), \\ z(A) &= (z_1(A), z_2(A), \dots, z_q(A)). \end{aligned}$$

令 \mathbf{I}_q 为 q 阶单位阵, $a = (a_1, a_2, \dots, a_q)^\top$ 为下界约束值构成的向量, 再记 $\mathbf{1}$ 表示元素全为1的 q 维向量. 则两种坐标间的变换关系为

$$x(A) = Dz(A), \quad (4)$$

其中 $D = (1 - \sum_{i=1}^q a_i) \mathbf{I}_q + a \otimes \mathbf{1}^\top$. 若 D 非退化, 则(4)式的逆变换为

$$z(A) = D^{-1}x(A) = \frac{1}{R}(x(A) - a), \quad (5)$$

其中 $R = 1 - \sum_{i=1}^q a_i > 0$.

由此可见, 经过线性变换后 S^{q-1} 中任意一点 A' 在 S_a^{q-1} 中有唯一与之对应的点.

下面我们构造一种特殊的具有下界约束的区域. 设 $\tau = (x_1, x_2, \dots, x_q) \in S^{q-1}$ 是单纯形 S^{q-1} 中的一点, 取一个常数 $0 < a_0 < 1$, 记

$$a = (a_1, a_2, \dots, a_q)^\top, \quad a_i = (x_i - a_0)I(x_i > a_0), \quad i = 1, 2, \dots, q, \quad (6)$$

其中 $I(\cdot)$ 为示性函数.

此时显然有 $\sum_{i=1}^q a_i = \sum_{i=1}^q (x_i - a_0)I(x_i > a_0) < 1$, 即满足非退化条件. 令(3)中各个分量的下界约束为(6), 此时区域 S_a^{q-1} 是一个包含 τ 点 $q-1$ 维单纯形, 且点 τ 与 S_a^{q-1} 中各 $q-2$ 维平面的距离为 a_i , $i = 1, 2, \dots, q$. 特别的, 如果 $I(x_i - a_i) = 1$, $i = 1, 2, \dots, q$, 则 τ 点到 S_a^{q-1} 中各 $q-2$ 维平面的距离都是 a_0 , 此时 τ 是单纯形 S_a^{q-1} 的中心. 为方便叙述, 我们称 a_0 为下界约束量, a 为下界约束向量, S_a^{q-1} 是以 τ 为中心, 下界约束量为 a_0 的单纯形.

由3.1节中的指数逆变换的方法(1)或逆变换方法(2)的可以生成一个无附加约束的随机混料点集 $\mathcal{T} = \{\tau_1, \tau_2, \dots, \tau_{N_0}\} \subset S^{q-1}$, 将这些随机混料点按行排列成矩阵 $T = (\tau_1^\top, \tau_2^\top, \dots, \tau_{N_0}^\top)^\top = \{x_{ij}\}_{i,j=1}^{N_0,q}$, 再作线性变换, 令

$$T_a = TD^\top = T(qa_0 \mathbf{I}_q + \mathbf{1} \otimes a^\top).$$

记 $\mathcal{T}_a = \{\alpha_1, \alpha_2, \dots, \alpha_{N_0}\} \subset S_a^{q-1}$ 是由 T_a 的各行元素组成的点集, \mathcal{T}_a 中的元素与 \mathcal{T} 中的元素是一一对应的, 它是由 N_0 个无附加约束的随机混料点向 S_a^{q-1} 中的一一映射, 记

$$\mathcal{T}(\tau, N_0, a_0) = \{\tau\} \cup \mathcal{T}_a$$

是包含 τ 点以及 S_a^{q-1} 内的 N_0 个随机混料点的集合. 将 $\mathcal{T}(\tau, N_0, a_0)$ 中的点按行排列成矩阵, 记为

$$ME(\tau, N_0, a_0) = [\tau^\top, \alpha_1^\top, \alpha_2^\top, \dots, \alpha_{N_0}^\top]^\top. \quad (7)$$

再考虑附加约束, 令

$$\mathcal{T}_{\mathcal{D}_s}(\tau, N_0, a_0) = \{\tau\} \cup (\mathcal{T}_a \cap \mathcal{D}_s)$$

表示包含 τ 点及区域 $\mathcal{D}_s \cap S_a^{q-1}$ 内的随机混料点组成的集合. 设若 N_0 个随机混料点经过线性变换后有 n_0 ($n_0 \leq N_0$) 个落入 $\mathcal{D}_s \cap S_a^{q-1}$ 内, 将 $\mathcal{T}_{\mathcal{D}_s}(\tau, N_0, a_0)$ 中的点按行排列成矩阵, 记为

$$CE(\tau, N_0, a_0) = [\tau^\top, \alpha_{i_1}^\top, \alpha_{i_2}^\top, \dots, \alpha_{i_{n_0}}^\top]^\top. \quad (8)$$

例如在三分量混料系统中, 取 $\tau_1 = (0.6, 0.3, 0.1)$, $\tau_2 = (0.1, 0.1, 0.8)$, 生成两组包含 τ_1 , τ_2 的随机混料点集 $\mathcal{T}(\tau_1, 50, 0.1)$ 和 $\mathcal{T}(\tau_2, 150, 0.15)$, 如图1(a)所示. 在四分量混料系统

中, 取 $\tau_3 = (0.3, 0.2, 0.2, 0.3)$, $\tau_4 = (0, 0.1, 0.1, 0.8)$, 生成两组包含 τ_3 , τ_4 的随机混料点集 $\mathcal{T}(\tau_3, 500, 0.1)$ 和 $\mathcal{T}(\tau_4, 1500, 0.1)$, 如图1(b)所示.

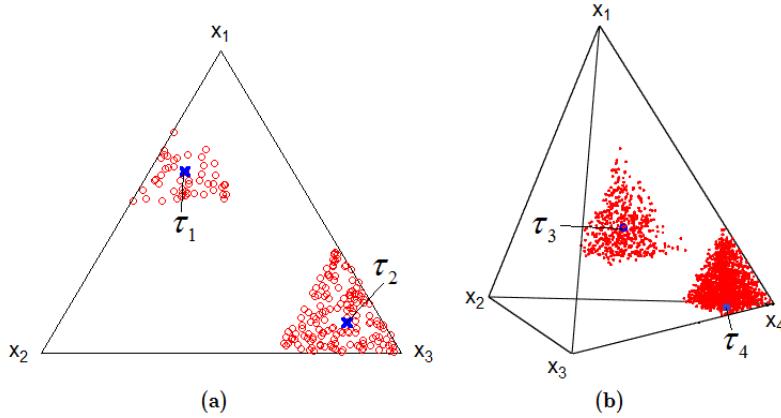


图1 (a) 三分量混料系统中两组包含 τ_1 , τ_2 的随机点集 $\mathcal{T}(\tau_1, 50, 0.1)$ 和 $\mathcal{T}(\tau_2, 150, 0.15)$
(b) 四分量混料系统中包含 τ_3 , τ_4 的随机点集 $\mathcal{T}(\tau_3, 500, 0.1)$ 和 $\mathcal{T}(\tau_4, 1500, 0.1)$

需要注意的是我们定义的矩阵(8)除了第一行 τ 是固定的之外, 其余各行都是相互独立的随机混料点, 并且都服从单纯形 S_a^{q-1} 上的均匀分布 $U(S_a^{q-1})$. 另外, 因为最终落入区域 \mathcal{D}_s 内的随机混料点个数不确定, 所以该矩阵的行数也是不确定的. 当 N_0 很大时, 能落入 \mathcal{D}_s 内的点数 n_0 应有以下关系

$$n_0 \approx N_0 \frac{\text{Vol}(\mathcal{D}_s \cap S_a^{q-1})}{\text{Vol}(S_a^{q-1})}.$$

如果 $\mathcal{D}_s \cap S_a^{q-1} \neq \emptyset$, 我们总能够通过增大 N_0 的方法产生足够多的随机混料点使之落入区域 $\mathcal{D}_s \cap S_a^{q-1}$ 内.

§4. MDRS算法的迭代搜索过程

为获得在具有复杂约束的试验域 \mathcal{D}_s 上的D-最优设计, 我们需要确定一组点集 $\mathcal{T}^* = \{\tau_1^*, \tau_2^*, \dots, \tau_k^*\}$ 和一个测度向量 $w = (w_1, w_2, \dots, w_k)$, 使得在试验域 \mathcal{D}_s 上的设计

$$\xi^* = \begin{pmatrix} \tau_1^* & \tau_2^* & \dots & \tau_k^* \\ w_1 & w_2 & \dots & w_k \end{pmatrix}$$

是D-最优或是近似D-最优的.

为此, 本文首先考虑在等测度条件下, 先确定一组点集, 然后再计算各点的测度, 最后由G-A等价定理来检验设计是否为D-最优. 在4.1节和4.2节计算信息阵行列式时, 我们总是令 $\text{diag}(w_1, w_2, \dots, w_k) = \mathbf{I}_k$.

这里先给出几个记号, 设矩阵 $A = \{a_{ij}\}_{i,j=1}^{n,m}$ 的各行元素为 $a_i = (a_{i1}, a_{i2}, \dots, a_{im})$, $i = 1, 2, \dots, n$. 记 $A[i] = a_i$ 是矩阵 A 的第 i 行, 记 $A[-i] = [a_1^\top, a_2^\top, \dots, a_{i-1}^\top, a_{i+1}^\top, \dots, a_n^\top]^\top$ 表示将 A 的 i 行剔除后所得到的 $(n-1) \times m$ 阶矩阵. 设有 m 维向量 $b = (b_1, b_2, \dots, b_m)$, 定义

$$\mathcal{R}\{b, A[-i]\} = [a_1^\top, \dots, a_{i-1}^\top, b^T, a_{i+1}^\top, \dots, a_n^\top]^\top,$$

表示将 A 的第 i 行替换为向量 b .

下面分两种情况讨论MDRS算法的随机搜索过程.

4.1 全点搜索

如果已知的回归模型中未知参数不太多, 我们可以考虑在每步随机搜索中让 k 个点同时进行搜索, 具体步骤如下.

首先按照3.1节中的方法生成 k 点 $\mathcal{T}^{(0)} = \{\tau_1^{(0)}, \tau_2^{(0)}, \dots, \tau_k^{(0)}\} \subset \mathcal{D}_s$ 作为迭代初值, 给定正整数 N_0 和下界约束量的初值 $a^{(0)}$. 将 k 个点代入(8)从而生成 k 个矩阵, 形如:

$$C_i^{(0)} = CE(\tau_i, N_0, a^{(0)}), \quad i = 1, 2, \dots, k. \quad (9)$$

设矩阵 $C_i^{(0)}$ 中共有 $N_i^{(0)}$ 行, $C_i^{(0)}$ 是由 τ 和 $S_{a^{(0)}}^{q-1} \cap \mathcal{D}_s$ 内的 $N_i^{(0)} - 1$ 个随机混料点按行排列而成的矩阵, 以 $C_i^{(0)}[j]$ 表示 $C_i^{(0)}$ 的第 j 行, 再从 $C_1^{(0)}, C_2^{(0)}, \dots, C_k^{(0)}$ 中各取出一行, 组成 k 行的矩阵

$$Z_l = \begin{bmatrix} C_1^{(0)}[i_1] \\ C_2^{(0)}[i_2] \\ \vdots \\ C_k^{(0)}[i_k] \end{bmatrix}, \quad i_j = 1, 2, \dots, N_i^{(0)}, \quad j = 1, 2, \dots, k, \\ l = 1, 2, \dots, \prod_{i=1}^k N_i^{(0)}. \quad (10)$$

令 $N^{(0)} = \prod_{i=1}^k N_i^{(0)}$, 这样, 各个矩阵 Z_l ($l = 1, 2, \dots, N^{(0)}$) 对应的设计记为 $\xi_l^{(0)}$ ($l = 1, 2, \dots, N^{(0)}$), 计算出所有设计对应的信息阵行列式, 并令

$$\xi^{(1)} = \arg \max_{1 \leq l \leq N^{(0)}} \det M(\xi_l^{(0)}).$$

记 $\xi^{(1)}$ 所对应的设计点为 $\mathcal{T}^{(1)} = \{\tau_1^{(1)}, \tau_2^{(1)}, \dots, \tau_k^{(1)}\}$, 即完成一次搜索迭代. 重复以上过程若干次, 最终得到一组点集. 以上迭代过程可以归纳为如下步骤:

步骤1 产生初值 $\mathcal{T}^{(0)} = \{\tau_1^{(0)}, \tau_2^{(0)}, \dots, \tau_k^{(0)}\} \subset \mathcal{D}_s$;

步骤2 $j = 0$, 产生 k 个矩阵 $C_i^{(j)} = CE(\tau_i^{(j)}, N_0, a^{(j)})$, $i = 1, 2, \dots, k$. 并由此产生 $N^{(j)}$ 个矩阵 $Z_l^{(j)}$, $l = 1, 2, \dots, N^{(0)}$ 如(10)所示;

步骤3 $\xi_{\max}^{(j)} = \arg \max_{1 \leq l \leq N^{(j)}} \det M(\xi_l^{(j)})$, $\mathcal{T}^{(j+1)} = \{\tilde{\tau}_1^{(j)}, \tilde{\tau}_2^{(j)}, \dots, \tilde{\tau}_k^{(j)}\}$.

循环步骤2和步骤3, 并令 $j = j + 1$, $\xi^{(j+1)} = \xi_{\max}^{(j)}$, 给定一个很小的正数 $\varepsilon > 0$, 令 $a^{(j+1)} = a^{(j)} - (a^{(0)} - \varepsilon(k\sqrt{q(q-1)})^{-1})/M$, 迭代至第 M 步停止迭代. 下面通过一个实例来确定椭球区域内的近似D-最优设计.

例1 将三种成分的原料聚乙烯 x_1 、聚苯乙烯 x_2 以及聚苯烯 x_3 混合在一起制成纤维, 并纺成纱线. 在此试验中, 试验者感兴趣的是影响纱线的伸长度, 即在固定作用力下, 一定长度的纱线的伸长是多少, 根据实践经验, 试验者感兴趣的领域可近似取为

$$0.27 \leq x_1 \leq 0.59, \quad 0.15 \leq x_2 \leq 0.45, \quad 0.2 \leq x_3 \leq 0.34, \quad (11)$$

对具有上、下界约束的领域稍加修改, 变成一个椭球邻域, 形为

$$\left(\frac{x_1 - 0.43}{0.16}\right)^2 + \left(\frac{x_2 - 0.3}{0.15}\right)^2 + \left(\frac{x_3 - 0.27}{0.07}\right)^2 \leq 1.$$

此椭球的中心在现行操作混料点 $\tau_1 = (0.43, 0.3, 0.27)$, 而0.16, 0.15及0.07分别表示 x_1 , x_2 , x_3 变程的一半. 下面通过全点搜索的方法, 确定一个以三阶Scheffé中心多项式模型

$$y = \sum_{i=1}^3 \beta_i x_i + \sum_{i < j} \beta_{ij} x_i x_j + \beta_{123} x_1 x_2 x_3$$

为回归模型下的7点设计.

通过3.1节中的方法产生一组初值, 每步搜索以各迭代点为中心产生 $N_0 = 4$ 个随机混料点, 令 $M = 1400$, $a^{(0)} = 0.08$, $\varepsilon = 0.0001$, 这样共进行1400步迭代, 最终信息阵的行列式值为 $\det M(\xi^{(1400)}) = 6.8489e-23$. 最终确定的7个设计点分别为 $\tau_1 = (0.544, 0.216, 0.240)$, $\tau_2 = (0.317, 0.380, 0.303)$, $\tau_3 = (0.504, 0.191, 0.305)$, $\tau_4 = (0.393, 0.271, 0.337)$, $\tau_5 = (0.467, 0.330, 0.203)$, $\tau_6 = (0.353, 0.410, 0.236)$, $\tau_7 = (0.43, 0.3, 0.27)$; 设计点集及迭代过程信息阵行列式的变化情况如图2所示.

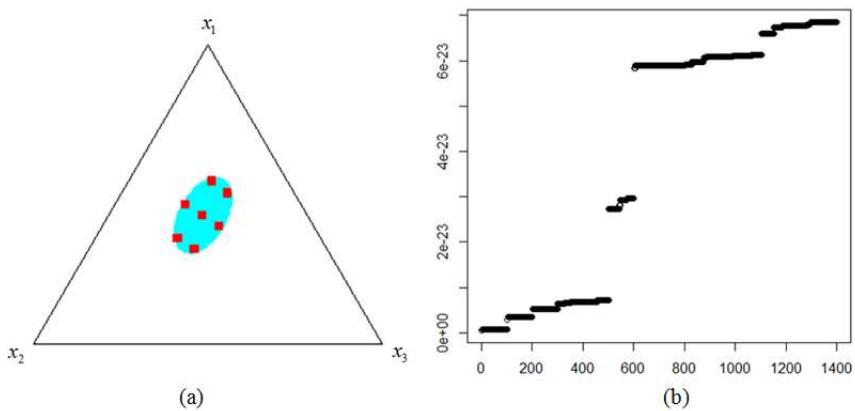


图2 (a) 最终确定的7个设计点 (b) 通过1400次迭代信息阵的行列式值散点图

如果本例中采用极端顶点设计,由附加约束(11)产生6个极端顶点,分别是 $\tau'_1 = (0.35, 0.45, 0.20)$, $\tau'_2 = (0.51, 0.15, 0.34)$, $\tau'_3 = (0.59, 0.21, 0.20)$, $\tau'_4 = (0.27, 0.39, 0.34)$, $\tau'_5 = (0.59, 0.15, 0.26)$, $\tau'_6 = (0.27, 0.45, 0.28)$,在此基础上添加一个中心点 $\tau_7 = (0.43, 0.3, 0.27)$ 构成7点设计,这时会发现这个设计的信息阵 $M(\xi')$ 是退化的,它的含有一个很小的特征值 $\lambda = 2.542234e-15$.这里说明两点:

(i) 全点搜索仅针对未知参数个数较少的情形,如果设置每步迭代搜索产生 N_0 个随机混料点,那么一次迭代可能需要对 N_0^k 个结果进行比较,当 N_0 较大,计算量会急剧增加,不易实施,如果 N_0 较小又不能保证存在能使 $\det M(\xi)$ 增大的点集.通过试验表明,在 $N_0^k \leq 10^5$ 时进行全点搜索是可行的.

(ii) 给定一个很小的常数 $\varepsilon > 0$,定义第 M 步的下界约束量为 $a^{(M)} = \varepsilon/[k\sqrt{q(q-1)}$,则下界约束量每一步以步长 $(a^{(M)} - a^{(0)})/M$ 缩小,它能保证迭代进行到第 M 步时有

$$\sum_{i=1}^k d(\tau_i^{(j+1)}, \mathcal{T}^{(j)}) \leq \varepsilon$$

成立,在迭代过程中,下界约束量逐步缩小,从而搜索范围也逐渐缩小,以此防止过多随机点未落入 $\mathcal{D}_s \cap S_a^{q-1}$ 内.

图3是方差函数 $\delta(\tau, \xi)$ 在约束区域内的等高图,从图中可见,方差函数在椭圆的边界区域达到7,但是经过计算,在边界还存在点使得方差函数略大于7,我们所得到的结果虽然不是D-最优的,但方差函数最大值与7差异不大,可以作为一个近似D-最优解.

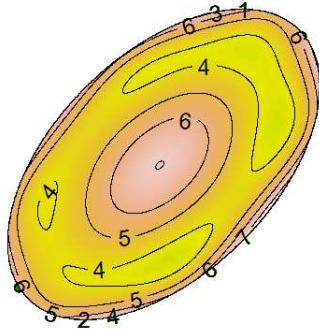


图3 \mathcal{D}_s 内方差函数的等高图

4.2 逐点搜索

如前文所述,当模型未知参数个较多时,使用全点搜索的计算量会非常大.例如对于四分量二阶Scheffé多项式中包含10个未知参数,对应的饱和设计需要取10个设计点,如果在 $CE(\tau_i, N_0, a^{(0)})$ 中每步产生 $N_0 = 10$ 个随机点,那么一步迭代需要比较 11^{10} 个矩阵的行列式值,这是不易实施的.

为此, 我们考虑采用逐点随机搜索的方法. 首先由3.1节中的方法产生一组初始点集, 将每个点按行排列为矩阵 $T^{(0)} = [(\tau_1^{(0)})^\top, (\tau_2^{(0)})^\top, \dots, (\tau_k^{(0)})^\top]^\top$. 设矩阵

$$C_i^{(j)} = CE(\tau_i^{(j)}, N_0, a^{(j)}), \quad i = 1, 2, \dots, k$$

共有 $N_i^{(j)}$ 行.

逐点搜索是将初始点集 $\mathcal{T}^{(0)} = \{\tau_1^{(0)}, \tau_2^{(0)}, \dots, \tau_k^{(0)}\}$ 中 $\tau_1^{(0)}$ 进行一步搜索, 将各 $C_1^{(0)}[i]$, $i = 1, 2, \dots, N_1^{(0)}$ 与 $T^{(0)}[-1]$ 合并, 令

$$Z_{1i}^{(0)} = \mathcal{R}\{C_1^{(0)}[i], T^{(0)}[-1]\}, \quad i = 1, 2, \dots, N_1^{(0)}. \quad (12)$$

各矩阵 $Z_{1i}^{(0)}$, $i = 1, 2, \dots, N_1^{(0)}$ 都是 $k \times q$ 阶矩阵, 除第一行不同外, 其余各行都相同, 这些矩阵所对应的设计为 $\xi_{1i}^{(0)}$, $i = 1, 2, \dots, N_1^{(0)}$, 计算出这些的信息阵行列式, 并选出最大值, 记为

$$\xi_1^{(0)} = \arg \max_{1 \leq l \leq N_1^{(0)}} \det M(\xi_{1i}^{(0)}).$$

$\xi_1^{(0)}$ 所对应的点集为 $\mathcal{T}_1^{(0)} = \{\tau_1^{(1)}, \tau_2^{(0)}, \dots, \tau_k^{(0)}\}$, 用 $\mathcal{T}_1^{(0)}$ 替换 $\mathcal{T}^{(0)}$, 将该点集中个点按行排成矩阵 $T_1^{(0)}$ 再进行下一步迭代. 这一过程称为一步搜索.

下面再对 $\tau_2^{(0)}$ 进行一步搜索, 令

$$Z_{2i}^{(0)} = \mathcal{R}\{C_2^{(0)}[i], T_1^{(0)}[-2]\}, \quad i = 1, 2, \dots, N_2^{(0)}.$$

类似的, 计算出所有设计对应的信息阵行列式并选出最大者, 进而得到一组点集 $\mathcal{T}_2^{(0)} = \{\tau_1^{(1)}, \tau_2^{(1)}, \dots, \tau_k^{(0)}\}$. 以此类推, 直到对 k 个设计点都进行了一步搜索, 则称此过程为一轮搜索. 第一轮搜索所确定的搜索点集为

$$\mathcal{T}^{(1)} = \mathcal{T}_k^{(0)} = \{\tau_1^{(1)}, \tau_2^{(1)}, \dots, \tau_k^{(1)}\}.$$

重复 M 轮搜索, 最终得到一组点集 $\mathcal{T}^{(M)} = \mathcal{T}_k^{(M-1)} = \{\tau_1^{(M)}, \tau_2^{(M)}, \dots, \tau_k^{(M)}\}$. 以上迭代过程可以归纳为如下步骤:

步骤1 产生初值 $\mathcal{T}^{(0)} = \{\tau_1^{(0)}, \tau_2^{(0)}, \dots, \tau_k^{(0)}\} \subset \mathcal{D}_s$;

步骤2 $i = 1, j = 0$, 产生 $N_i^{(j)}$ 个矩阵 $C_i^{(j)} = CE(\tau_i^{(j)}, N_0, a^{(j)})$. 并由此产生 $N_i^{(j)}$ 个矩阵

$$Z_{il}^{(j)} = \mathcal{R}\{C_i^{(j)}[l], T_{i-1}^{(j)}[-l]\}, \quad l = 1, 2, \dots, N_i^{(j)};$$

步骤3 令 $\xi_i^{(j)} = \arg \max_{1 \leq l \leq N_i^{(j)}} \det M(\xi_{il}^{(j-1)})$, $\mathcal{T}_i^{(j)} = \{\tau_1^{(j)}, \dots, \tau_i^{(j)}, \tau_{i+1}^{(j-1)}, \dots, \tau_k^{(j-1)}\}$.

给定下界约束量初值 $a^{(0)}$ 和一个很小的正数 ε , 令 $a^{(j)} = a^{(j-1)} - (a^{(0)} - \varepsilon)/M$. 令 $j = j + 1$, 当 $j \geq k$ 时, $i = i + 1$, 且 $\mathcal{T}^{(j)} = \mathcal{T}_k^{(j-1)}$, 当 $i \geq M$ 时停止.

对于 k 点设计, 逐点随机搜索的每一步搜索至多比较 N_0 个结果, 一轮搜索共需要比较 kN_0 个结果, 比全点搜索大大节省了内存, 并且经过实验, 逐点随机搜索的结果也能达到近似D-最优. 由D-最优准则可知, 如果设计是饱和设计, 在每点的测度应相等, 对于非饱和设计, 每个设计点处的测度有可能不相等, 这就需要确定各个设计点处的测度. 下面讨论在设计点确定的情形下再确定各点测度的方法.

4.3 测度的确定

设若按照4.2节中的方法最终确定了一组点集 $\mathcal{T}^* = \{\tau_1^*, \tau_2^*, \dots, \tau_k^*\}$, 我们也通过随机搜索的方法来确定最优测度. 设 $w^{(0)} = (w_1^{(0)}, w_2^{(0)}, \dots, w_k^{(0)})$ 是一个 k 维行向量, 以 $w^{(0)}$ 作为随机搜索的初值, 令

$$\xi^{(0)} = \begin{pmatrix} \mathcal{T}^* \\ w^{(0)} \end{pmatrix} = \begin{pmatrix} \tau_1^* & \tau_2^* & \cdots & \tau_k^* \\ w_1^{(0)} & w_2^{(0)} & \cdots & w_k^{(0)} \end{pmatrix}$$

作为初始设计.

然后在每步随机搜索中, 固定 \mathcal{T}^* 对测度向量进行随机搜索. 具体可按以下步骤进行.

步骤1 令 $i = 0$, 由(7)定义 $M^{(i)} = ME(w^{(i)}, N_0, a^{(i)})$, 设 $M^{(i)}$ 的各行元素分别为 $M^{(i)}[j]$, $j = 1, 2, \dots, N_0$. 再记

$$\xi^{(i)} = \begin{pmatrix} \mathcal{T}^* \\ w^{(i)}[j] \end{pmatrix};$$

步骤2 令 $i = i + 1$, $a^{(i+1)} = a^{(i)} - M^{-1}(a^{(0)} - \varepsilon)$, $\xi^{(i+1)} = \arg \max_{1 \leq j \leq N^{(i)}} \det M(\xi_j^{(i)})$, 其中 $\xi^{(i+1)}$ 所对应的测度向量记为 $w^{(i+1)}$.

重复以上步骤 M 次, 最终确定的设计的测度为向量 $w^{(M)} = (w_1^{(M)}, w_2^{(M)}, \dots, w_k^{(M)})$. 显然, 每一步搜索, 对于设计的信息阵行列式值都是单调不减的. 且由于下界约束量的控制, 能使得测度向量最终收敛到一个很小的范围内. 当我们最终确定了一个 k 点设计的设计点集 \mathcal{T}^* 和它所对应的测度, 就可以用G-A等价定理来检验设计 ξ 是否为D-最优的. 下面我们以一个实例来说明逐点搜索的有效性. 在此引用文献[8]中的例1的试验域进行计算.

例 2 三分量混料系统中, 考虑二阶Becker回归模型

$$\begin{aligned} H_1 : y &= \sum_{i=1}^3 \beta_i x_i + \sum_{i < j} \beta_{ij} \min(x_i, x_j) \\ H_2 : y &= \sum_{i=1}^3 \beta_i x_i + \sum_{i < j} \frac{\beta_{ij} x_i x_j}{x_i + x_j} \\ H_3 : y &= \sum_{i=1}^3 \beta_i x_i + \sum_{i < j} \beta_{ij} (x_i x_j)^{1/2} \end{aligned} \tag{13}$$

在区域 $\mathcal{D}_s = \{(x_1, x_2, x_3) : \sum_{i=1}^3 x_i = 1, x_i \geq 0, x_1^2 + x_2^2 \leq 0.36\}$ 的渐近饱和D-最优设计.

对于以上三个模型进行逐点随机搜索, 最初考虑每个模型都使用饱和设计, 但是发现如果采用饱和设计, 每个模型都无法达到D-最优. 为此, 对各个模型进行了调整, 第一个模型采用7点设计, 后两个模型采用8点设计, 每步搜索产生100个下界约束随机混料点, 共进行100轮随机搜索. 表1是使用MDRS算法经过逐点随机搜索后得到的最终结果.

表1 \mathcal{D}_s 内Becker模型的近似D-最优设计

模型 H_1				模型 H_2				模型 H_3			
x_1	x_2	x_3	w	x_1	x_2	x_3	w	x_1	x_2	x_3	w
0	0	1	0.171	0	0.6	0.4	0.13576	0	0.182	0.818	0.09347
0.5	0	0.5	0.148	0.292	0.524	0.184	0.13293	0	0.001	0.999	0.14618
0.333	0.334	0.333	0.159	0.538	0.265	0.197	0.12797	0.268	0.537	0.195	0.11935
0.568	0.193	0.239	0.134	0.247	0.247	0.506	0.13540	0.225	0	0.775	0.11007
0	0.5	0.5	0.153	0.6	0	0.4	0.13458	0.6	0	0.4	0.14643
0.425	0.423	0.152	0.104	0	0.329	0.671	0.08607	0	0.6	0.4	0.14221
0.202	0.565	0.233	0.131	0	0	1	0.15828	0.536	0.269	0.195	0.11001
				0.334	0	0.666	0.08902	0.169	0.175	0.656	0.13229
$\det M(\xi^{(700)}) = 2.370336e-09$				$\det M(\xi^{(800)}) = 2.348032e-11$				$\det M(\xi^{(800)}) = 2.162175e-09$			

在模型 H_2 中的设计点 $(0, 0, 1)$ 是一个近似值, 在计算逆矩阵时会产生无效项(因为 $x_1 x_2 / (x_1 + x_2)$ 一项会出现 $0/0$ 的形式), 信息阵的行列式值是将该点替换为 $(0, 0.001, 0.999)$ 再进行计算所得. 从以上结果中可以发现, 不论哪个模型, 其中都包含了顶点 $(0, 0, 1)$, 模型 H_2 和 H_3 都含有 $(0.6, 0, 0.4)$ 和 $(0, 0.6, 0.4)$. 所以在设计之初, 对模型类型无法断定时, 可以使用MDRS算法对多个模型搜索D-最优设计, 最终选择多个模型最优点集中都包含的设计点.

图4–图6分别是模型 H_1 , H_2 , H_3 的结果, 其中(a)是最终确定的设计点, (b)是迭代过程中信息阵行列式值的散点图, 横轴为迭代搜索的次数, 纵轴为信息阵行列式值, (c)是对应模型的方差函数的曲面图. 从下列图中(c)可见, 方差函数的曲面都是在6下方, 并且在边界处达到6, 这说明所确定的设计是近似D-最优的.

§5. 加点试验与边界搜索

在实际生产生活中会遇到这样的情形, 试验者对假定的回归模型提出了设计, 但是实际拟合效果不佳, 这时需要对模型进行调整, 或更换模型类型, 或改变设计点. 为了原有的数据不被浪费, 考虑更换模型后, 在已确定的设计点基础上添加若干个设计点, 再次进行试验. 那么所添加的设计及其测度都需要确定.

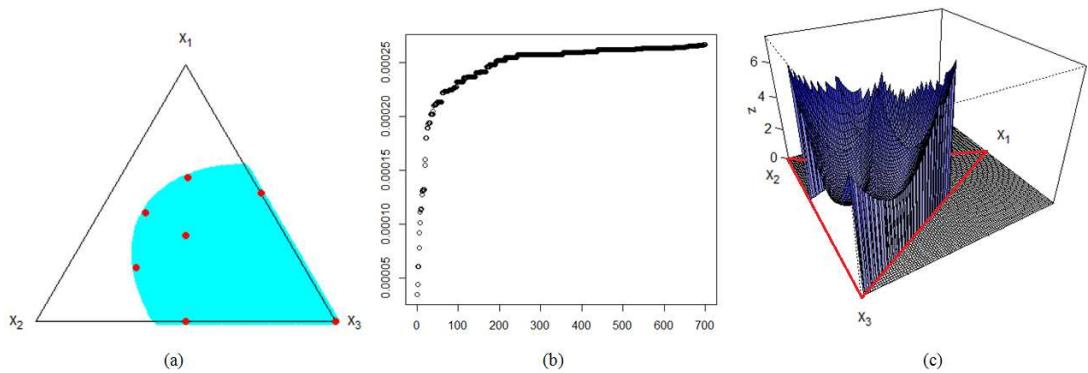


图4 模型 H_1 的结果

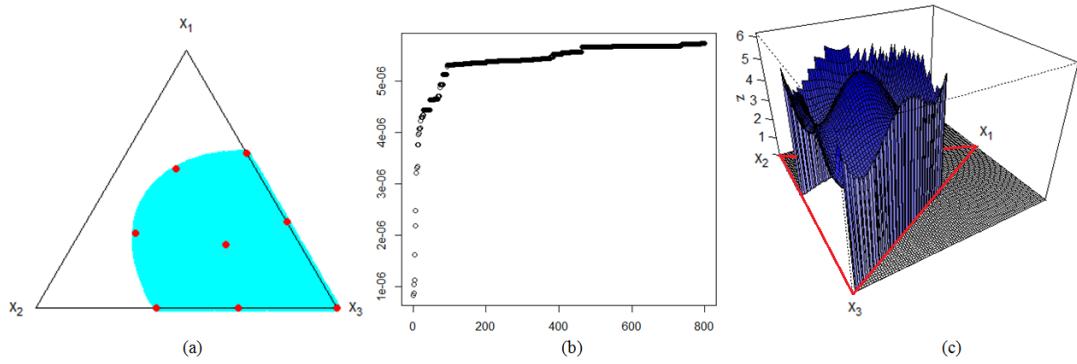


图5 模型 H_2 的结果

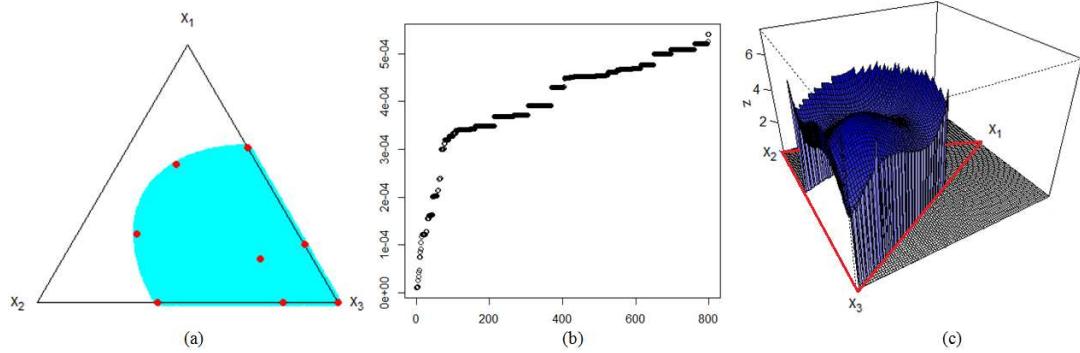


图6 模型 H_3 的结果

我们考虑在原有设计点集 \mathcal{T} 的基础上添加 p 个试验点，使得设计在新的模型下达到近似D-最优。同样的，也可以使用随机搜索的方法确定需要添加的点集。

首先从 $\mathcal{S} = \{s_1, s_2, \dots, s_N\} \subset \mathcal{D}_s$ 中选出若干个Bootstrap样本, 每组样本包含 p 个点, 按行排列成 N_0 个矩阵, 与已确定的 k 点设计 ξ 的矩阵为 T 合并, 从所有设计中选出信息阵行列式最大者, 令 $\mathcal{S}^{(0)} = \{s_1^{(0)}, s_2^{(0)}, \dots, s_p^{(0)}\} \subset \mathcal{D}_s$, 并以此作为迭代初值. 然后再应用MDRS算法进行搜索. 具体可按以下步骤进行.

步骤1 产生初值 $\mathcal{S}^{(0)} = \{s_1^{(0)}, s_2^{(0)}, \dots, s_p^{(0)}\} \subset \mathcal{D}_s$;

步骤2 $i = 1, j = 0$, 产生 $N_i^{(j)}$ 个矩阵 $C_i^{(j)} = CE(\tau_i^{(j)}, N_0, a^{(j)})$. 并由此产生 $N_i^{(j)}$ 个矩阵

$$Z_{il}^{(j)} = \mathcal{R}\{C_i^{(j)}[l], S_{i-1}^{(j)}[-l]\}, \quad l = 1, 2, \dots, N_i^{(j)}.$$

令 $V_{il}^{(j)} = [(Z_{il}^{(j)})^\top, T^\top]^\top$, 所对应的设计为 $\xi_{il}^{(j)}$, $l = 1, 2, \dots, N_i^{(j)}$;

步骤3 令 $\xi_i^{(j)} = \arg \max_{1 \leq l \leq N_i^{(j)}} \det M(\xi_{il}^{(j-1)})$, $\mathcal{S}_i^{(j)} = \{s_1^{(j)}, \dots, s_i^{(j)}, s_{i+1}^{(j-1)}, \dots, s_k^{(j-1)}\}$.

给定下界约束量初值 $a^{(0)}$ 和一个很小的正数 ε , 令 $a^{(j)} = a^{(j-1)} - (a^{(0)} - \varepsilon)/M$. 令 $j = j + 1$, 当 $j \geq p$ 时, $i = i + 1, j = 0$, 且 $\mathcal{S}^{(j)} = \mathcal{S}_k^{(j-1)}$, 当 $i \geq M$ 时停止.

添加的设计点, 不论是全点搜索还是逐点搜索都有一个共同之处, 都是将原有的设计点矩阵固定, 再与随机点矩阵合并, 以此进行迭代, 直到结果收敛到一定范围.

通过若干轮搜索后, 起初分散的设计点绝大部分点会“重合”或“近似重合”在几个点处. 一般的, 对于给定的很小的常数 $\varepsilon > 0$, 在迭代过程中如果有 $d(\tau_i, \tau_j) \leq \varepsilon$, 则认为 τ_i 与 τ_j 是重合的. 例如我们考虑在某约束区域 \mathcal{D}_s 内产生 100 个初始设计点, 通过若干轮迭代后会发现这 100 个点最终都重合到几个点上, 这说明在 D- 最优准则下, 我们只需要取定几个“重要”的设计点即可. 对多点进行随机搜索, 最后需要将重合的点合并, 合并原则为: 如果添加点与原设计点重合, 取原设计点, 添加点与添加点重合, 取 D- 准则下更优者.

要将重合点合并, 可以按照以下方法进行: 设所得到的结果是由 $k + p$ 个点按行排列的矩阵 T , 记 $\tau_i, i = 1, 2, \dots, k + p$ 是矩阵 T 的第 i 行元素. 取定一个很小的正数 $\varepsilon > 0$, 再定义一个 $k + p$ 阶方阵 E , E 中包含主对角线的上三角部分中的各元素定义为 $I_{ij} = I(d(\tau_i, \tau_j) < \varepsilon)$. E 中不包含主对角线的下三角部分元素全部令为 0. 然后再对 E 中的元素进行调整. 具体为: 如果 $\sum_{j=1}^{k+p} I_{ij} > 1$, 则令 E 的第 i 行元素全为 0, 如果 $\sum_{j=1}^{k+p} I_{ij} = 1$, 则 E 的第 i 行元素不变. 我们

把调整后的矩阵记为 $G = \{g_{ij}\}_{i,j=1}^{k+p}$, 最后所得的矩阵中有若干行为 0. 令 $r_i = \sum_{j=1}^{k+p} g_{ij}, i = 1, 2, \dots, k + p$ 就表示 τ_i 点以及与 τ_i 点重合的点的个数. 将向量 $r = (r_1, r_2, \dots, r_{k+p})^\top$ 与 T 合并为 (T, r) 后, 将 $r = 0$ 的行删除即可剔除重复的点.

以例 1 中模型 H_1 为例, 如果我们开始考虑的是饱和设计, 在 6 点设计的基础上添加 p 个设计点, 记为 $\mathcal{S} = \{s_1, s_2, \dots, s_p\} \subset \mathcal{D}_s$, 在此, 我们考虑了三种情况: 即添加点的个数分别为 $p_1 = 24, p_2 = 48, p_3 = 96$, 最终确定的设计点的位置与图 7 完全相同. 说明所添加的设计点基本与已有的设计点重合.

图 7 中的 9 个设计点分别为: $\tau_1 = (0.6, 0, 0.4), \tau_2 = (0, 0, 1), \tau_3 = (0.423, 0.425, 0.152), \tau_4 = (0, 0.6, 0.4), \tau_5 = (0.221, 0.558, 0.221), \tau_6 = (0, 0.5, 0.5), \tau_7 = (0.5, 0, 0.5), \tau_8 = (0.557, 0.221, 0.222), \tau_9 = (0.333, 0.333, 0.333)$. 由 G-A 等价定理可以验证: 在等测度条件下的 9 点设计是 D- 最优的.

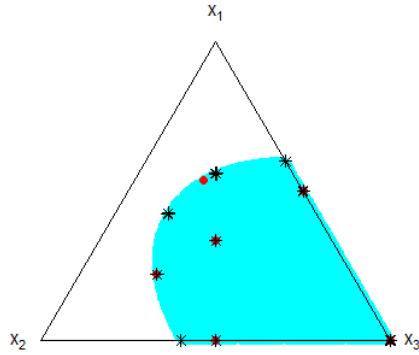


图7 图中红色方点为原有的设计点，“*”为添加的设计点

由于D-最优准则所取定的设计点一般分布在单纯形的边界及顶点上，在约束区域 \mathcal{D}_s 内也是类似的，例1和例2中，设计点都分布在约束区域的边界上。如果为了使得D-最优准则下搜索到的结果更为精确，可以考虑在边界上进行随机搜索。

一般的，线性约束区域的边界是 $q-2$ 维的平面，而非线性约束的边界则是 $q-2$ 维曲面。对于 $\tau \in S^{q-1}$ ，约束边界可以表示为 $\eta(\tau) = 0$ 的形式，对于与约束边界附近的一点 τ' ，如果它与边界很近，即对于一个很小的正数 $\varepsilon > 0$ ，有 $d(\tau', \mathcal{C}) \leq \varepsilon$ 成立，其中 $\mathcal{C} = \{\tau : \tau \in S^{q-1}, \eta(\tau) = 0\}$ 。若把 \mathcal{C} 中与 τ' 最近的一点记为 $\tau_0 = (x_{01}, x_{02}, \dots, x_{0q})$ ，我们希望能产生以 τ_0 为中心，在约束边界上的随机混料点。

首先，给定一个常数 $c > 0$ ，令 $a_1 = \max\{x_{01} - c, 0\}$, $b_1 = \min\{x_{01} + c, 1\}$ ，将各个分量定义为均匀分布的随机点，令 $x_1 \sim U(a_1, b_1)$ ，再令

$$a_i = \max\{x_{0i} - c, 0\}, \quad b_i = \min \left\{ \max \left\{ x_{0i} + c, 1 - \sum_{j=1}^{i-1} x_j \right\}, 1 \right\}, \\ x_i \sim U(a_i, b_i), \quad i = 2, 3, \dots, q-2.$$

这样， τ' 的前 $q-2$ 个分量都是服从均匀分布的随机变量，剩下的两个分量由边界约束和基本约束决定，即由 $\eta(x_{q-1}, x_q; x_1, x_2, \dots, x_{q-2}) = 0$ 和 $\sum_{i=1}^q x_i = 1$ 解出后两个分量。

由以上步骤生成的 N 个随机点 $\tau'_1, \tau'_2, \dots, \tau'_N$ 及 τ_0 组成的集合记为 $\mathcal{T}_\eta(\tau_0, N, a) = \{\tau_0\} \cup \{\tau'_1, \tau'_2, \dots, \tau'_N\}$ ，类似于第3节中的讨论，我们定义一个约束边界上的点集矩阵： $LE_\eta(\tau_0, N, a)$ 是约束边界 η 上的随机点集 $\mathcal{T}_\eta(\tau_0, N, a)$ 按行排列的矩阵。我们称 a 是边界约束量。

同样的，可以使用MDRS算法在边界上进行随机搜索。例如图7中，我们考虑在三个边界 $\eta_1(\tau) = x_1 = 0$, $\eta_2(\tau) = x_2 = 0$ 上各产生两个随机点，在 $\eta_3(\tau) = x_1^2 + x_2^2 = 0.36$ 上产生三个随机点，固定中心点和顶点，在边界上选定初值后，与固定点排列成矩阵，对7个点进行搜索，按照逐点搜索的方法，每一步搜索是从其中一个边界上的随机点集阵 $LE_\eta(\tau, N, a)$ 种取出一行，替换掉上一步所确定的矩阵中的某一行，每步替换一行，逐行替换，并规定边界约束量 a 逐渐变小，直到搜索区域缩小到一定范围。边界搜索基本思想与第4节中所讨论是一

样的, 在此就不重复赘述了.

§6. 结 论

本文使用MDRS算法在具有复杂约束的混料试验域内搜索D-最优设计点集, 并确定设计点的测度, 通过实例验证了该方法的有效性. 除此之外, 我们用这种方法研究了其他模型在不同的约束区域内的D-最优设计, 包括多项式模型、可加混料模型、倒数模型、齐次模型, 以及这些模型混合所得的模型等等. 通过MDSR算法都能找到D-最优, 或近似D-最优的设计点集及对应的测度, 限于篇幅, 此处就没有一一讨论. 值得一提的是, SA算法理论上是依概率收敛于全局最优解的, 但本文构造的MDRS算法并没有像SA算法那样, SA算法每一步搜索都按一定概率不选择当前最优解, 而MDRS算法中每步搜索后的 $\det(M(\xi))$ 都是单调非降的. 在此说明两点:

(i) 在约束区域较小, 方差函数与信息矩阵行列式都会出现多极值的情形, 每步迭代都保证信息矩阵行列式单调不减, 在搜索区域内逐步调整, 最终能确定一个近似D最优的设计.

(ii) 每步迭代产生具有下界约束区域内的随机点集, 下界约束量初值适当取大一些. 例如选择约束区域最大直径的 $1/q$, 然后随着迭代进行, 下界约束量缓缓变小, 以此缩小搜索区域. 每一步迭代都有可能落入局部最优解, 而下一步迭代都有一定的概率跳出局部最优解, 也就是说每一步迭代都有机会“弥补”上一步迭代的错误选择. 一般的, 下界约束量越大, 且产生的随机点越多, 跳出局部最优的概率也就越大, 所以在逐点搜索中, 我们设置的产生随机点数往往比全点搜索要多得多, 并且运算时间也是可以接受的. 由图2中(b)可以发现, 在许多步迭代中 $\det(M(\xi))$ 是没有发生变化的, 直到下界约束量小到一定程度时才有上升的趋势, 由此可见很多步迭代其实是无效的, 但是我们又不能让下界约束量缩小得太快, 那样会增大落入局部最优解的概率. 鉴于此, 我们可以尝试设置门限参数 $t(j)$, 每步迭代只有当 $\det(M(\xi^{(j+1)})) - \det(M(\xi^{(j)})) > t(j)$ 时才接受新解, 否则重新产生随机点, 再进行比较. 门限接受的随机搜索实际上是随机搜索的一种压缩, 它能提高效率, 使得更快达到最优结果, 但是门限参数的设置需要更进一步的确定.

总之, MDRS算法简单易行, 并且随机搜索的结果可以进一步调试, 这样可能使得结果更加逼近最优设计点集, 然后固定点集, 对测度向量进行优化进而求解出满足D-最优准则下的测度. 该算法能在短时间内为试验者提供一系列适用于不同模型在复杂约束下的近似D-最优的设计方案.

参 考 文 献

- [1] Scheffé H. Experiments with mixtures [J]. *J. Roy. Statist. Soc. Ser. B*, 1958, **20**(2): 344–360.

- [2] Cornell J A, Good I J. The mixture problem for categorized components [J]. *J. Amer. Statist. Assoc.*, 1970, **65(329)**: 339–355.
- [3] Wang Y, Fang K T. Uniform design of experiments with mixtures [J]. *Sci. China Ser. A*, 1996, **39(3)**: 264–275.
- [4] Chan L Y. Optimal designs for experiments with mixtures: a survey [J]. *Comm. Statist. Theory Methods*, 2000, **29(9-10)**: 2281–2312.
- [5] Wang Y, Fang K T. Number theoretic methods in applied statistics (II) [J]. *Chinese Ann. Math. Ser. B*, 1990, **11(3)**: 384–394.
- [6] Borkowski J J, Piepel G F. Uniform designs for highly constrained mixture experiments [J]. *J. Qual. Technol.*, 2009, **41(1)**: 35–47.
- [7] Deng W L, Wang J L. Uniform design of mixture experiments with constraints [J]. *J. East China Norm. Univ. Natur. Sci. Ed.*, 2002, **2002(2)**: 26–32.
- [8] Liu Y, Liu M Q. Construction of uniform designs for mixture experiments with complex constraints [J]. *Comm. Statist. Theory Methods*, 2016, **45(8)**: 2172–2180.
- [9] Fang K T, Wang Y. *Number-Theoretic Methods in Statistics* [M]. London: Chapman and Hall, 1994.
- [10] Fang K T, Yuan K H, Benter P M. Quantizers and representative points of elliptically contoured distributions [C] // *The First Conference on Recent Developments in Statistical Research*, December, 1990, Hong Kong.
- [11] 关颖男. 混料试验设计 [M]. 上海: 上海科学技术出版社, 1990.

Asymptotically D-Optimal Design of Mixture Experiment with Complex Constraints

LI GuangHui ZHANG ChongQi

(School of Economic and Statistic, Guangzhou University, Guangzhou, 510006, China)

Abstract: It is difficult to get an accurate optimum design when the experimental design area is very irregular under complex constraints. This paper constructs a random search algorithm for mixture experiments designed (MDRS). Firstly, generating an initial points set in areas with complex constraints by the Monte-Carlo method, then use MDRS algorithm iterative to approximate optimum set. By way of example verification, this method is effective. It can be used as a standard measure of other designs, that is the only effective when given superior to other designs approximate optimal solution.

Keywords: Monte-Carlo method; D-criterion; information matrix

2010 Mathematics Subject Classification: 62K05; 62K99